Experiment No.7
Perform DCL and TCL commands
Date of Performance:
Date of Submission:

Aim :- Write a query to implement Data Control Language(DCL) and Transaction Control Language(TCL) commands

Objective :- To learn DCL commands like Grant and Revoke privileges to the user and TCL commands to commit the transactions and recover it using rollback and save points.

Theory:

Data Control Language:

DCL commands are used to grant and take back authority from any database user.

- Grant
- Revoke

a. Grant: It is used to give user access privileges to a database.

Example

1. GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;

b. Revoke: It is used to take back permissions from the user.

Example

1. REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;

Transaction Control Language

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.

These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

Here are some commands that come under TCL:

- COMMIT
- ROLLBACK
- SAVEPOINT

a. Commit: Commit command is used to save all the transactions to the

database. Syntax:

1. COMMIT;

Example:

1. DELETE FROM CUSTOMERS
2. WHERE AGE = 25;
3. COMMIT;

b. Rollback: Rollback command is used to undo transactions that have not already been saved to the database.

Syntax:

1. ROLLBACK;

Example:

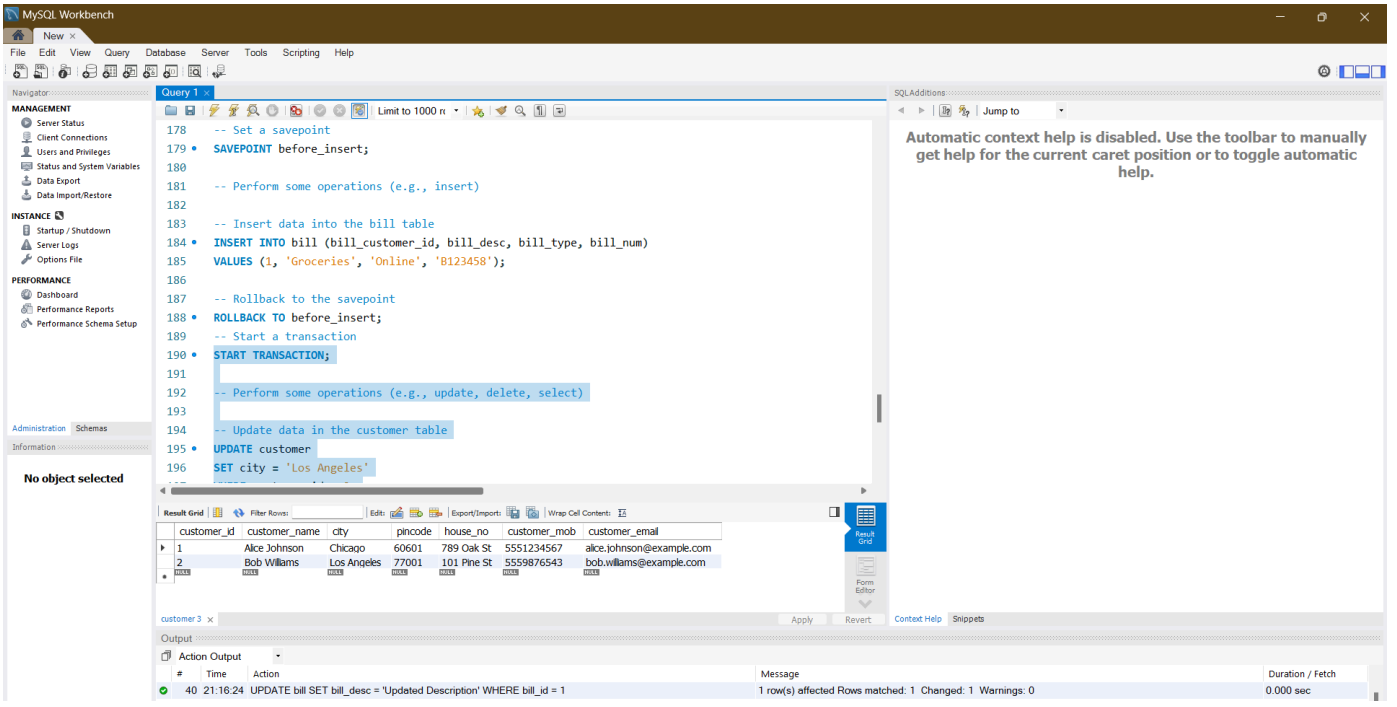
1. DELETE FROM CUSTOMERS
2. WHERE AGE = 25;
3. ROLLBACK;

c. SAVEPOINT: It is used to roll the transaction back to a certain point without rolling back the entire transaction.

Syntax:

2. SAVEPOINT SAVEPOINT_NAME;

Implementation:



Conclusion:

1. Explain about issues faced during rollback in mysql and how it got resolved.
2. Explain how to create a user in sql.

In conclusion, issues faced during rollback in MySQL primarily revolve around the potential loss of data consistency or integrity due to incomplete rollback operations, especially in scenarios involving complex transactions or concurrent access. Some common issues include:

1. Partial Rollbacks: In multi-statement transactions, if one of the statements fails, MySQL might not be able to rollback the entire transaction, leaving the database in an inconsistent state.

2. Deadlocks: Concurrent transactions might encounter deadlocks, where each transaction is waiting for the other to release a lock, leading to potential rollback issues and transaction failure.

3. Long Rollback Times: Large transactions or transactions involving many changes can result in long rollback times, impacting database performance and availability.

To address these issues, MySQL provides several mechanisms and best practices:

1. Use of Transactions: Utilize transactions to group related operations and ensure atomicity, consistency, isolation, and durability (ACID) properties.

2. Error Handling: Implement robust error handling mechanisms to handle exceptions and errors gracefully, ensuring proper rollback and recovery procedures are followed.

3. Transaction Management: Carefully manage transactions, keeping them short and ensuring they are properly committed or rolled back to minimize potential issues.

4. Deadlock Detection and Resolution: Configure MySQL to detect and resolve deadlocks automatically, or implement application-level deadlock detection and resolution mechanisms.

5. Transaction Isolation Levels: Choose appropriate transaction isolation levels to balance data consistency and concurrency requirements, minimizing the risk of conflicts and rollback issues.

Regarding creating a user in SQL, here's a general procedure:

1. Syntax:
   CREATE USER 'username'@'hostname' IDENTIFIED BY 'password';

2. Explanation:
   - `CREATE USER`: This command creates a new user in the MySQL database.
   - `'username'@'hostname'`: Specifies the username and the hostname from which the user is allowed to connect to the MySQL server. If the user can connect from any host, you can use `'username'@'%'`.
   - `IDENTIFIED BY 'password'`: Sets the password for the user account.

For example:
CREATE USER 'myuser'@'localhost' IDENTIFIED BY 'mypassword';

This concludes the experiment, highlighting the potential issues faced during rollback in MySQL and

providing insights into resolving them, along with an explanation of how to create a user in SQL.